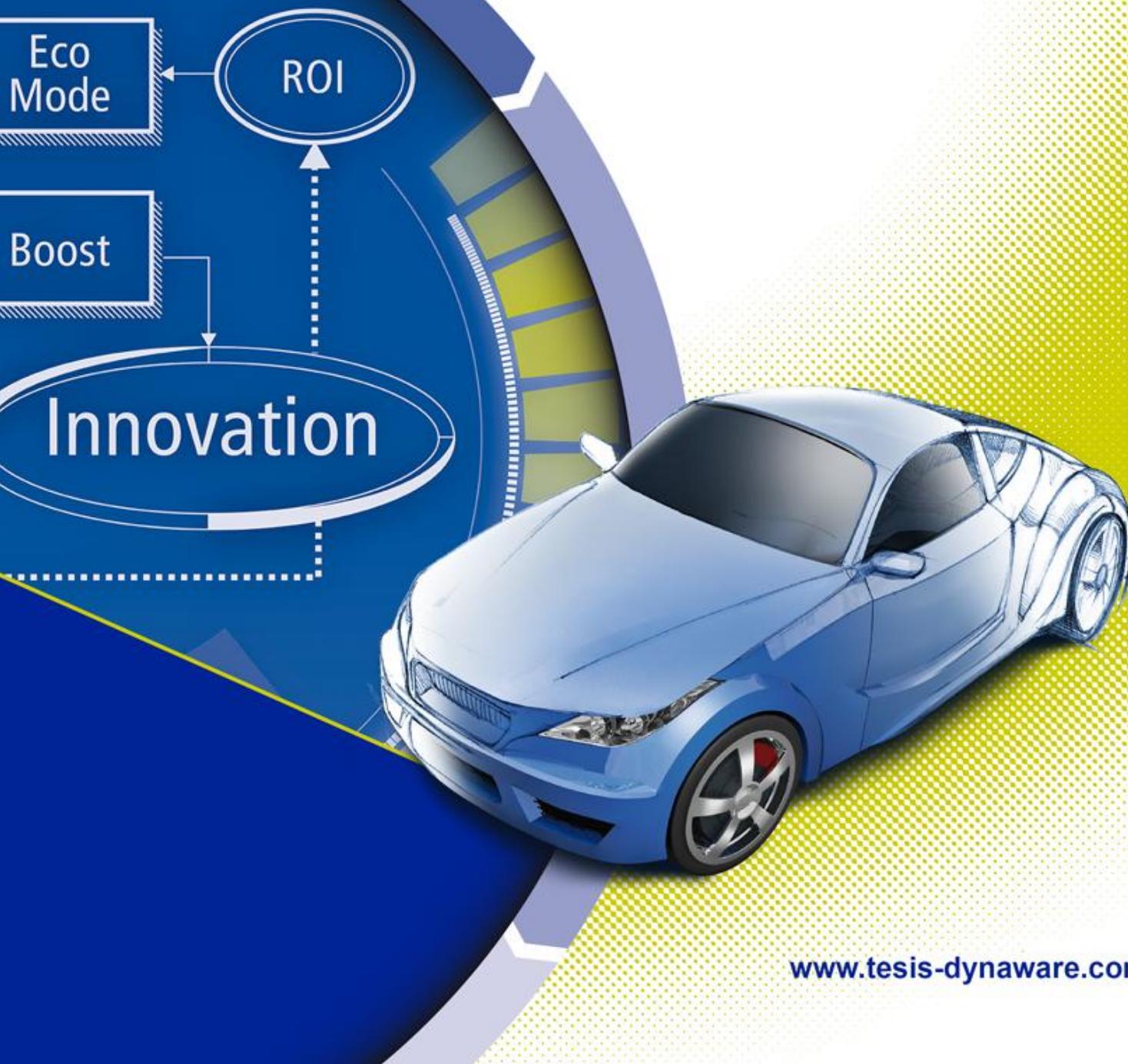


Importing Models from Physical Modeling Tools Using the FMI Standard



Overview

The objective of this tutorial is to demonstrate the workflow for the integration of FMUs in DYNA4.

The following use case is explained:

- The user has a simulation model from a physical modeling software tool, such as Dymola, AMESim or SimulationX that supports the tool independent Functional Mock-up Interface (FMI) standard.
- This model (for example a complete drivetrain, combustion engine or electric motor) is to be integrated as a Functional Mock-up Unit (FMU) in the overall model architecture that is a Simulink model with full DYNA4 integration.
- The model architecture shown in the tutorial is the one used by DYNA4 models, however the process works equally well with totally different user-specific model architectures.
- The FMI Toolbox for MATLAB from Modelon(<http://www.modelon.com/products/fmi-toolbox-for-matlab/>) is used as the interface to Simulink: Using DLL binaries for FMUs the simulation modes "Normal" and "Accelerator" can be used in Simulink.
- The simulation may be conducted as a co-simulation or the imported model may be simulated using the Simulink solver.

Tutorial contents:

- [Exporting Physical Models as FMUs](#): Describes the process of exporting FMUs from a physical modeling tool.
- [Importing FMUs into Simulink](#): Addresses the necessary setup steps for integrating FMUs into Simulink using the FMI Toolbox for MATLAB from Modelon.
- [Importing FMUs into DYNA4](#): Depicts the required steps to integrate FMUs into a functional DYNA4 simulation model based on Simulink.
- [Working with FMUs in DYNA4](#): Describes options for passing data to FMUs, tracing signals and simulating FMUs using Simulink and DYNA4.

FMI Toolbox compatibility:

- Restrictions regarding the supported physical modeling tools, that can be used for exporting FMUs, can be found on the Modelon documentation:
<http://www.modelon.com/products/fmi-toolbox-for-matlab/supported-tools/>.
- All MATLAB versions from R2007b onwards are supported by the FMI Toolbox for MATLAB. The User's Guide provides further information as to which versions have been tested:
http://www.modelon.com/fileadmin/user_upload/Products/Modelon/FMIT/FMIT-1_7-UsersGuide.pdf.
- Version 1.7 of the Modelon FMI Toolbox for MATLAB has been tested with DYNA4. All information provided in this tutorial is related to this version of the toolbox.

Exporting Physical Models as FMUs

The objective of this tutorial step is to show how to export models from physical simulation environments as FMUs. The following topics are addressed:

- Adjust the simulation model interface
- Export the simulation model as an FMU

Proceed as explained in the following:

1. Adjust the interface (ports) of your existing simulation model from a physical modeling tool. This must be done in a signal-based way so that the model can be exported as an FMU. This means input and output signals will be required to define this interface and it may be necessary to install sensors in order to measure certain model states and actuators in order to apply physical properties to the model.

Here we look at the example of a Dymola drivetrain model that we want to use as a new simulation component in our DYNA4 simulation model based on Simulink (see figure [Drivetrain model in Dymola](#)).

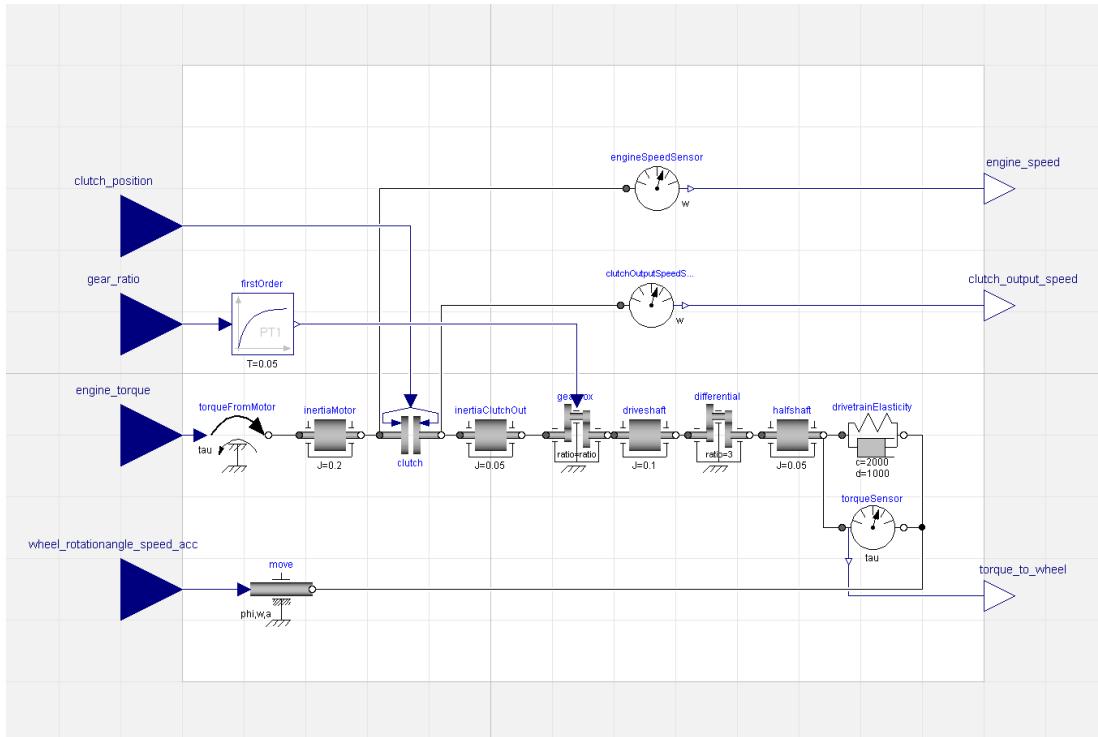


Figure: Drivetrain model in Dymola

The component engine of the DYNA4 simulation model should supply the engine torque (input signal "engine_torque") and needs to receive the engine speed signal (output signal "engine_speed") from the Dymola drivetrain model. This is achieved by using a torque source block as an actuator for the engine torque and a speed sensor block for the engine speed. The output torque acting on the wheel shafts (output signal "torque_to_wheel") is sent to the component wheel system of the DYNA4 simulation model, which will return the wheel's rotational angle, speed and acceleration (input signal "wheel_rotationangle_speed_acc"). In this example the motion of all driven wheels is combined in one wheel.

Normally, the Dymola model would work differently to solve the differential equations, but the use of the "move" block forces the system to use the input signal "wheel_rotationangle_speed_acc" as the rotational motion of the wheel. If for example the wheel system model was also included within the system boundary of the Dymola model, the interface signals would be different. It is advisable to choose the system boundaries with consideration of the resulting mathematical constraints.

2. Export the simulation model as an FMU by using the FMI export functionality of your physical simulation tool

In Dymola the options for the FMI export can be found in the Simulation Setup GUI (see figure [FMI export GUI in Dymola](#)).

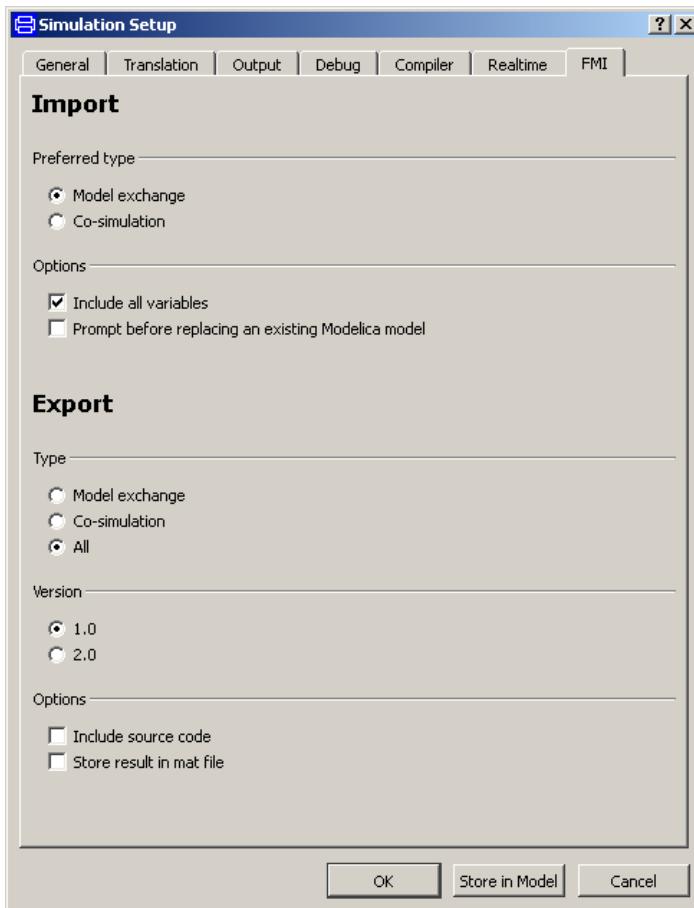


Figure: FMI export GUI in Dymola

There are three settings to be done: Select a **Type**, select an **FMI Version** and select some further **Options**.

As **Type** either **Model exchange** (FMI-ME) or **Co-simulation** (FMI-CS) can be selected. Model exchange exports the model without solver and Co-simulation exports an encapsulated model and solver.

As **Version** select 1.0 to ensure compatibility with V1.7 of the Modelon FMI Toolbox.

For the **Options** it is not necessary to include the source code or store result in mat file.

Importing FMUs into Simulink

The objective of this tutorial step is to show how to integrate an FMU into Simulink using the FMI

Toolbox for MATLAB from Modelon. The following topics are addressed:

- Install the FMI Toolbox for MATLAB from Modelon
- Access the FMU block in Simulink

Proceed as explained in the following:

1. Install the FMI Toolbox for MATLAB from Modelon and configure the MATLAB installation such that the FMI blockset libraries are available. Follow the installation instructions found on the Modelon website:
http://www.modelon.com/fileadmin/user_upload/Products/Modelon/FMIT/FMIT-1_7-UsersGuide.pdf
2. After installing and setup is complete, the FMI Toolbox should appear in the Simulink Library Browser (see figure FMI Toolbox in Simulink Library Browser).

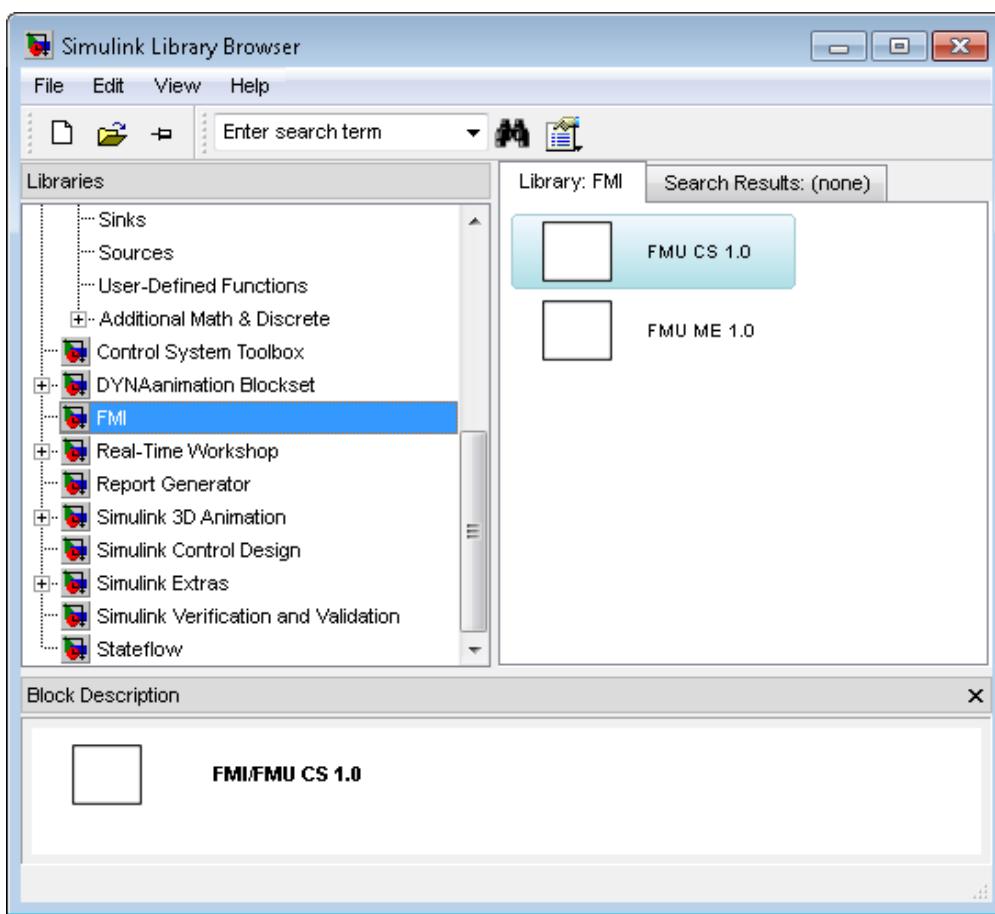


Figure: FMI Toolbox in Simulink Library Browser

3. Once this is available, the block can be used in any Simulink model. Drag and drop a copy of the required block into a blank Simulink model (see figure FMU block in Simulink). Either choose the block FMU ME for Model Exchange or FMU CS for Co-simulation.

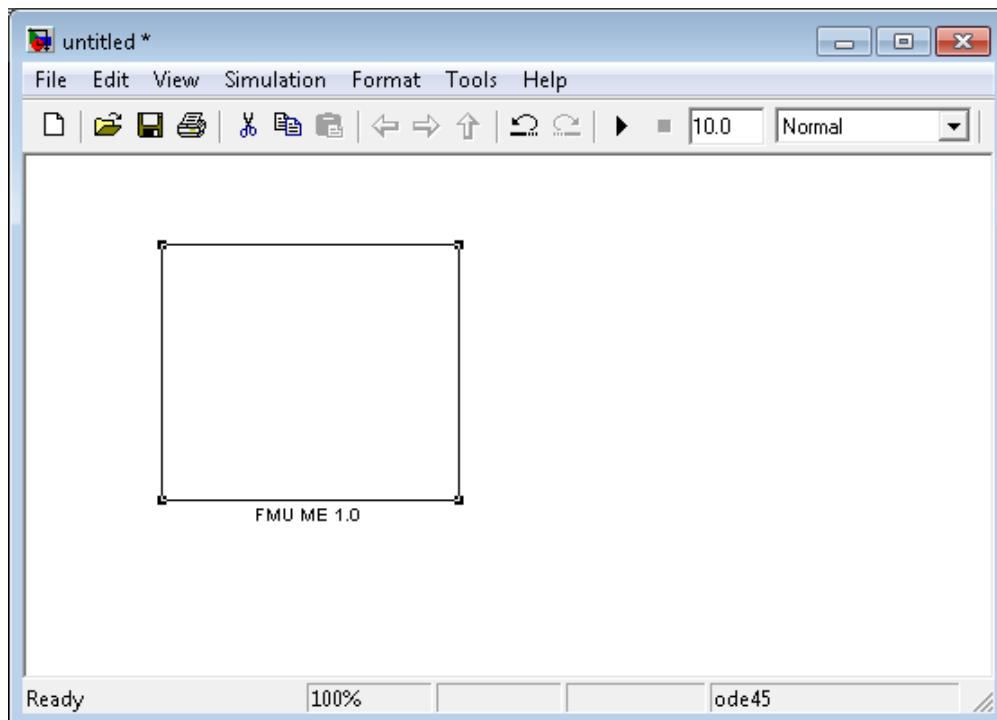


Figure: FMU Block in Simulink

Double clicking on the block opens a GUI, and the FMU file can be loaded. With the specifications from the loaded FMU file the FMU block automatically adjusts its ports including their names (see figure Specified FMU Block in Simulink).

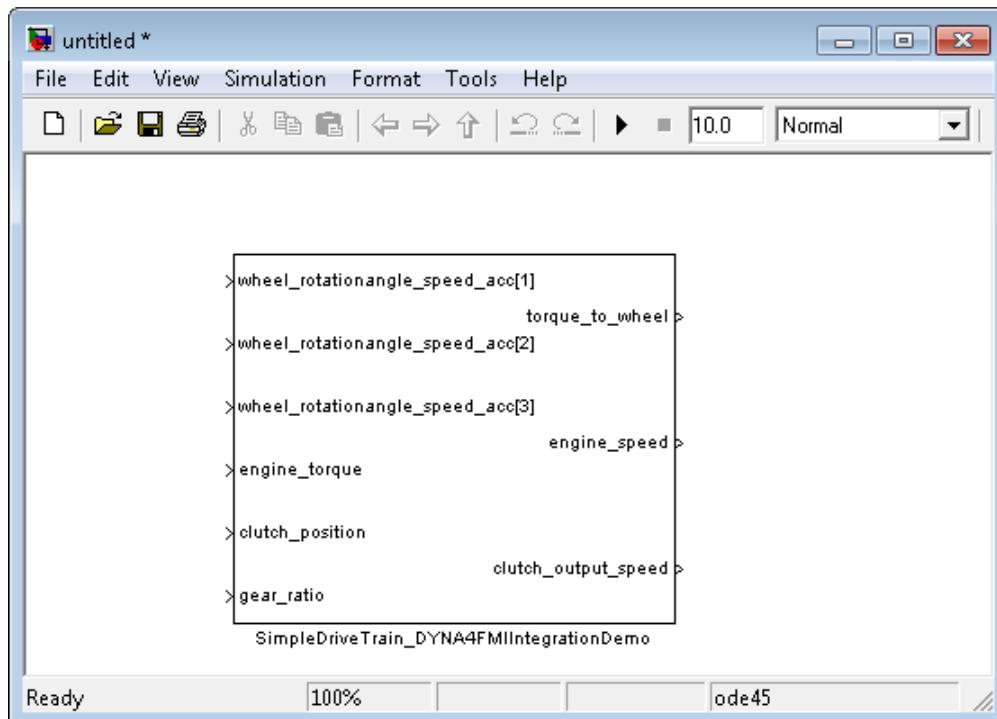


Figure: Specified FMU Block in Simulink

Importing FMUs into DYNA4

The objective of this tutorial step is to show how to integrate an FMU into the DYNA4 simulation framework. The following topics are addressed:

- Create a user-defined module including FMU blocks and adjust its settings
- Create a DYNA4 component including FMU modules and import this component into a simulation model

The following workflow is recommended because it allows maximum integration with the DYNA4 framework functionality.

The full integration process can be split into four main parts:

1. Create a user-defined module including FMU blocks.

This module will be saved in the DYNAdatabase, in a user-defined module library. In this way it can be used in any component model in DYNA4.

2. Adjust settings of FMU blocks in user-defined modules.

3. Create a DYNA4 component including FMU modules.

This component will be saved in the DYNAdatabase, as a reference DYNA4 component. This allows the user to import and use the component in any simulation project and any simulation model using the DYNA4 Replacing Model Components (for further information see Replacing Model Components in the DYNA4 User Manual) functionality.

4. Import an FMU component into a simulation model.

The final step is to import the reference component model from the DYNAdatabase into the desired simulation model.

Notes:

- Options regarding the methods of [Applying Data to FMU Blocks](#), tips regarding [Tracing Signals of FMU Blocks](#) and [Simulating with FMU Blocks](#) are discussed in the chapter [Working with FMUs in DYNA4](#).
- Before attempting to import an FMU block, it is helpful to have some understanding of the processes as described in the references, as they are very similar for importing a standard Simulink model without the FMU module.

Creating User-Defined Modules Including FMU Blocks

The objective of this tutorial step is to show how to create a user-defined module including FMU blocks. The following topics are addressed:

- Create a new module in the user-defined library in DYNA4
- Add an FMU block to this module
- Define block mask parameters for this module

Proceed as explained in the following:

1. Create a new user-defined module using the Manage Modules wizard (see figure Manage Modules page).

For detailed information on this topic refer to Creating a New User-Defined Module in the DYNA4 User Manual.

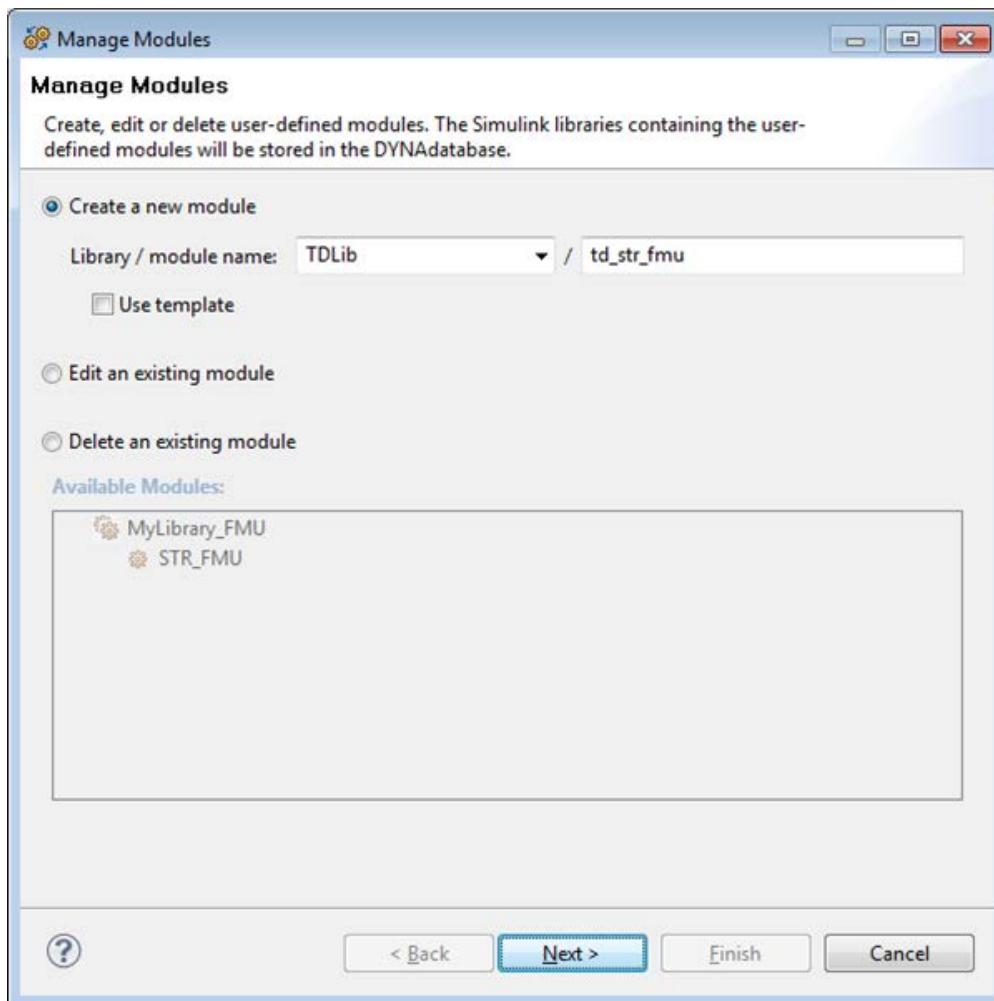


Figure: Manage Modules page

Click on the Next button without making any changes in Simulink, we will perform these later (see figure [Perform Changes in Simulink page](#)).

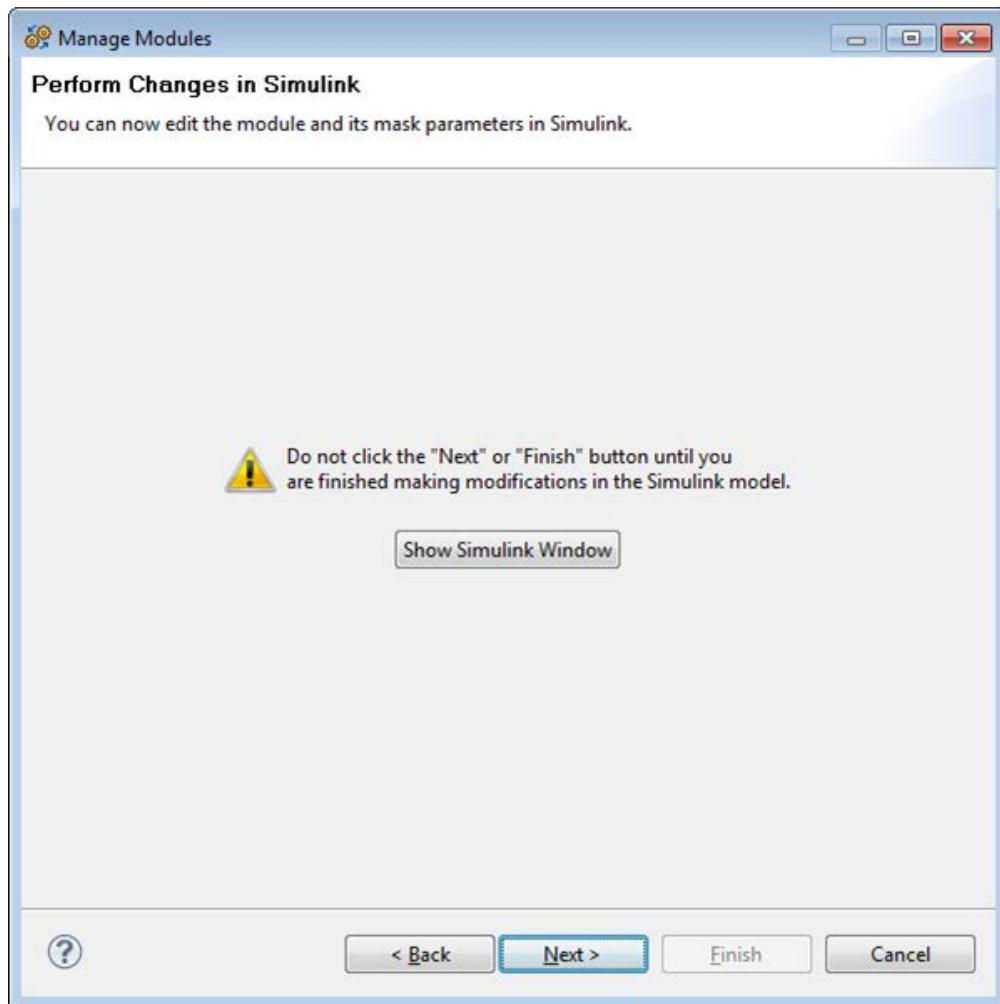


Figure: Perform Changes in Simulink page

Click on Finish without adding any parameters (see figure Edit Module Parameters page).

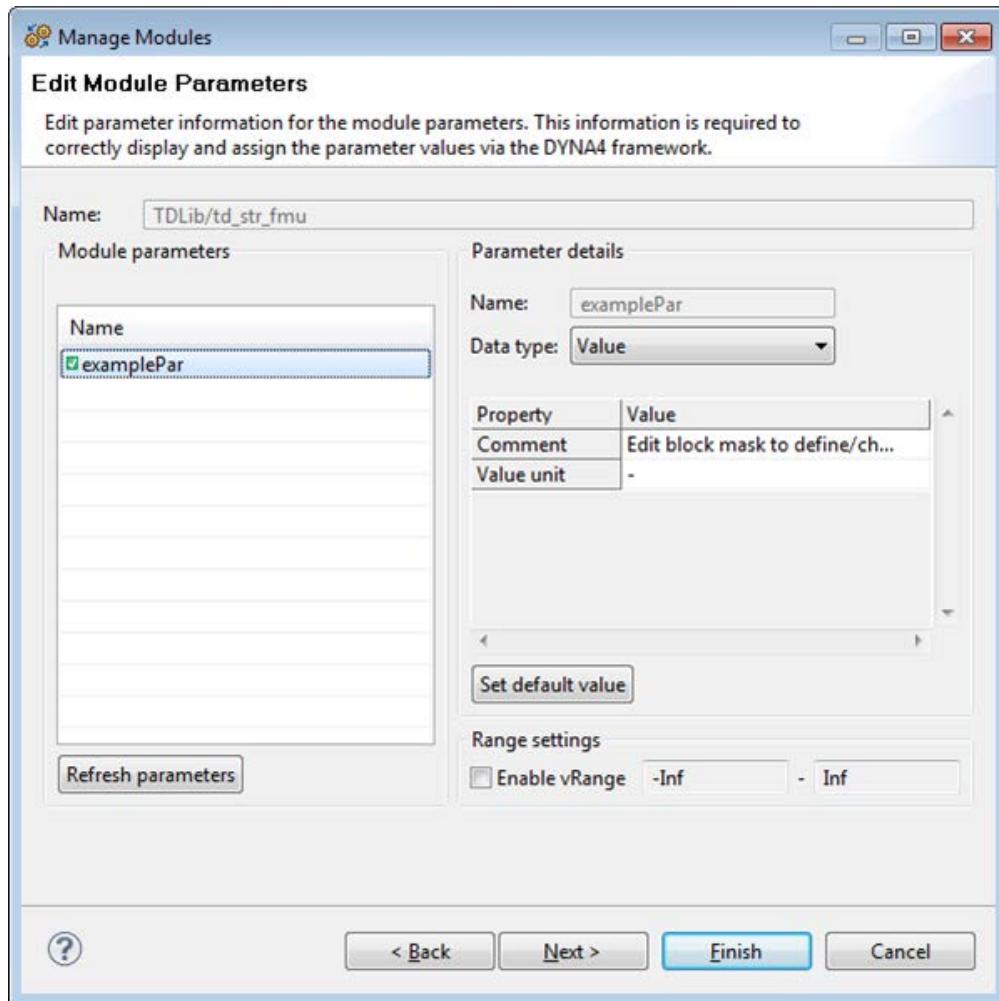


Figure: Edit Module Parameters page

Now the wizard has created an empty module and a folder in the DYNAdatabase.

2. Add the .fmu file to the user-defined module library directory (see figure [User-defined module library directory](#)).

Ensure that there are no other copies of the .fmu file on the MATLAB workspace.

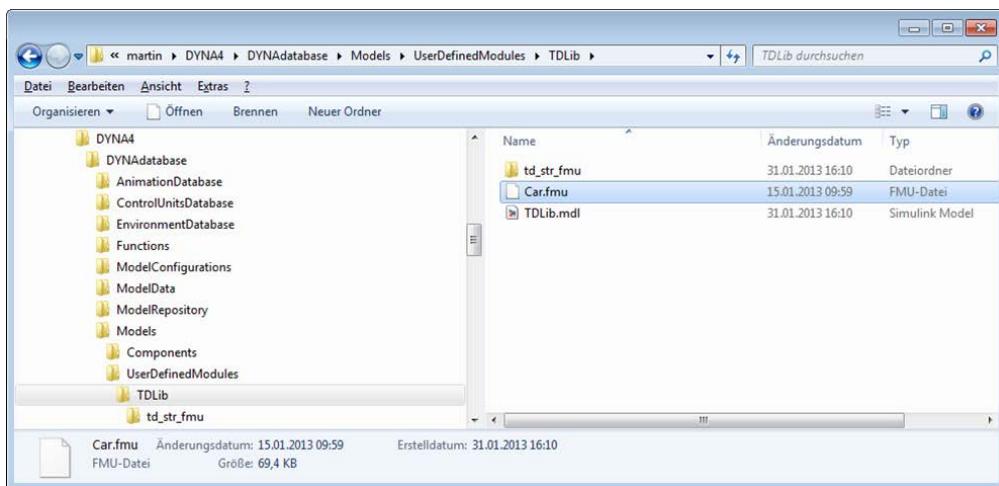


Figure: User-defined module library directory

3. Add the FMU block to the user-defined module library and define the Simulink block mask parameters (see figure [Mask editor and edit module parameters page](#)).

Therefore switch back to the Manage Modules wizard and follow the instructions in [Editing a User-Defined Modul](#) (for further information about [Editing a User-Defined Module](#) see the [DYNA4 User Manual](#)) and [Importing FMUs into Simulink](#). These parameters will then be recognized by the DYNA4 GUI and a corresponding data file template will be created.

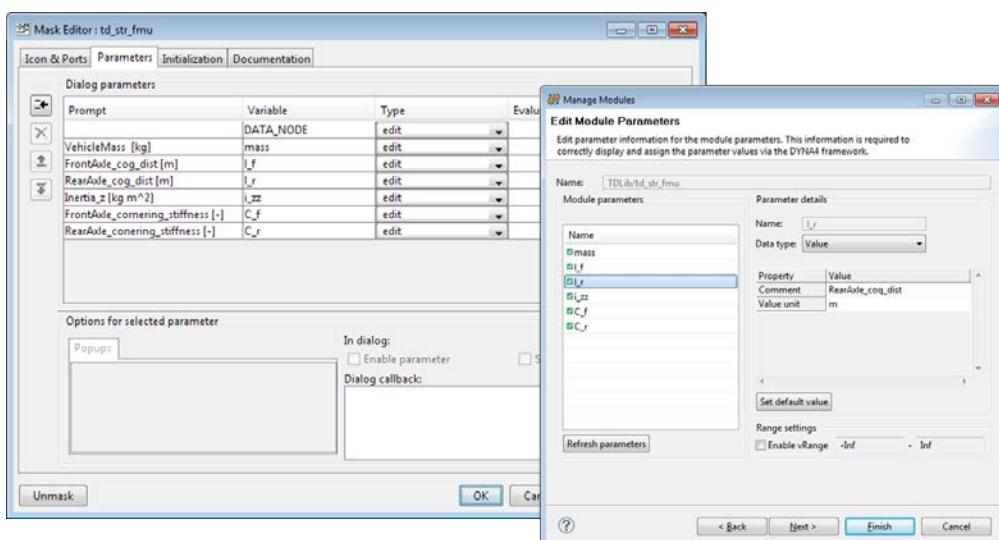


Figure: Mask Editor and Manage Module Parameters page

In order to use the parameter apply solution described in [Applying Data to FMU Blocks](#), the following convention should be used in the DYNA4 "Edit Module Parameters" GUI (see right window in figure [Mask Editor and Manage Module Parameters page](#)):

Parameter Name: May not contain the "." character. The "_" character is suggested e.g. "InertiaMotor_J".

Comment: Exact case sensitive name of the parameter as shown in the flat view of the FMI Toolbox GUI (see figure [Flat view of parameters in FMI Toolbox GUI](#)) e.g. "InertiaMotor.J".

This format also assists the user to quickly identify from the DYNA4 GUI exactly where each parameter is being applied in the model.

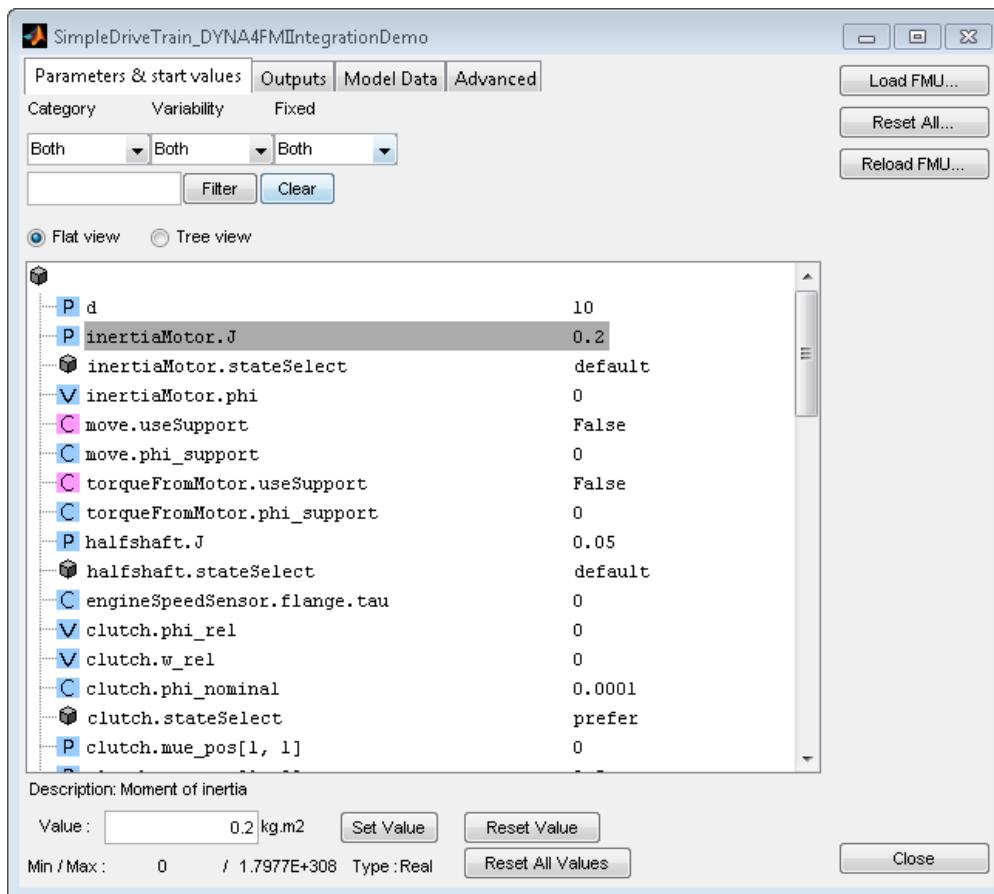


Figure: Flat view of parameters in FMI Toolbox GUI

Further settings that can be made in the "Edit Module Parameters" GUI (see right window in figure [Mask editor and edit module parameters page](#)) are:

- **Set default value:** Will create a default value that will be used whenever a new parameter file is created.
- **Range settings:** These should be used to ensure that only feasible parameters can be applied.

Notes:

- The DYNA4 Manage Modules wizard automatically creates a set of functions specific to the user-defined module which are called by the DYNA4 framework when certain events occur. When not required, they can be left blank. For further details please refer to Creating a New User-Defined Module in the DYNA4 User Manual.

Adjusting Settings of FMU Blocks in User-Defined Modules

The objective of this tutorial step is to give a short overview of the main settings that can be adjusted in the FMI Toolbox GUI. The following settings are addressed:

- Parameters and start values
- Outputs
- Advanced settings for FMU ME and FMU CS

Parameters and start values

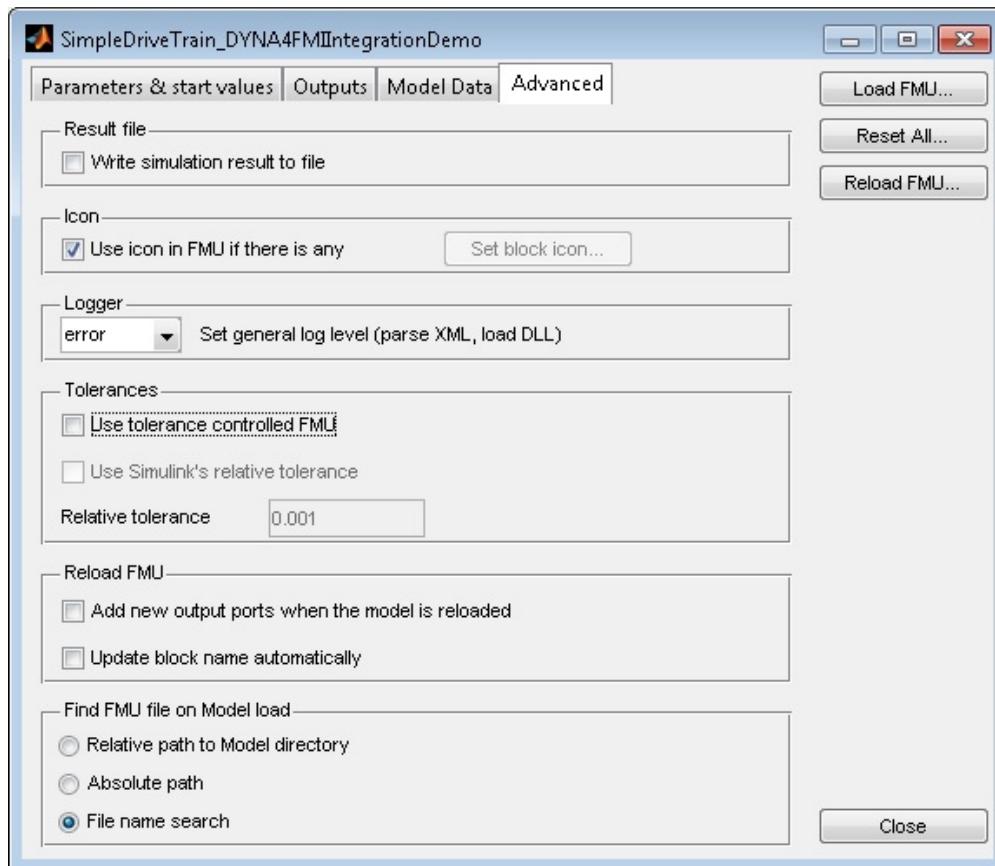
The default parameters and start values can be defined directly here.

Outputs

It is possible to select any constants or internal variables from the .fmu from the GUI and define these as output ports. This is necessary if they are to be traced using the DYNA4 trace mechanism. For further details refer to [Tracing Signals of FMU Blocks](#).

Advanced

The Advanced tab provides the opportunity to set various different options (see figure FMU Advanced tab).



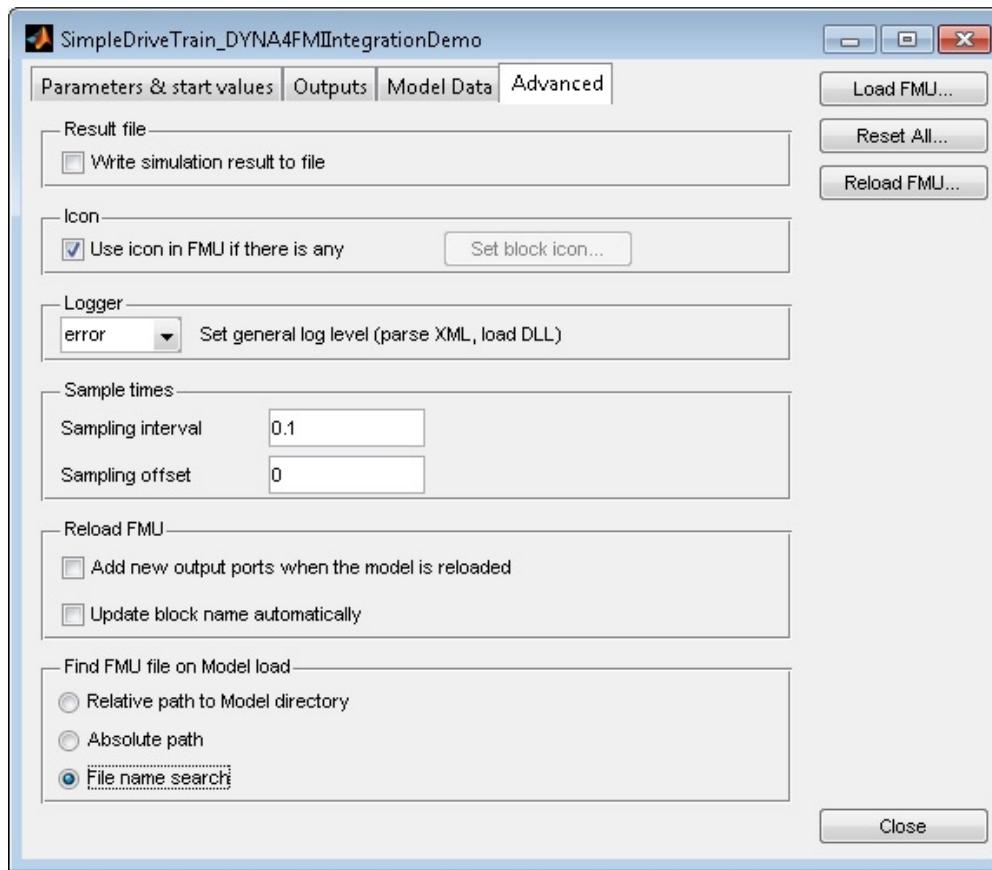


Figure: FMU Advanced tab

Result File: The Modelon Toolbox has the ability to create a report file of all internal states as a text document. The toolbox also provides MATLAB scripts which can be used to comfortably read this information into the MATLAB workspace, for postprocessing or plotting. This feature is discussed in more detail in [Tracing Signals of FMU Blocks](#).

Tolerances (FMU ME block only): Check that the tolerance control is turned off. The Simulink tolerance control is only relevant for simulink solvers with variable step size, and therefore is for DYNA4 Models which operate with ODE1 not to be used.

Sample times (FMU CS block only): Sampling interval and Sampling offset need to be defined here, since the FMU has its own solver.

Find FMU file on Model load: For an easily interchangeable FMU block the selection File name search is recommended.

Creating DYNA4 Components Including FMU Modules

The objective of this tutorial step is to show how to create a DYNA4 component from a user-defined FMU module. The following topics are addressed:

- Create a new DYNA4 component
- Integrate the user-defined FMU module into this component

Proceed as explained in the following:

1. Once the user-defined module has been defined the next step is to create a new component. This allows the new component to be directly imported into a simulation model. See Creating a New User-Defined Component Model in the DYNA4 User Manual for more details on how to create a DYNA4 component in general.
2. Integrate the user-defined FMU module into the new component (see figure Add FMU Modules page) and directly connect it to the rest of your simulation model in Simulink.

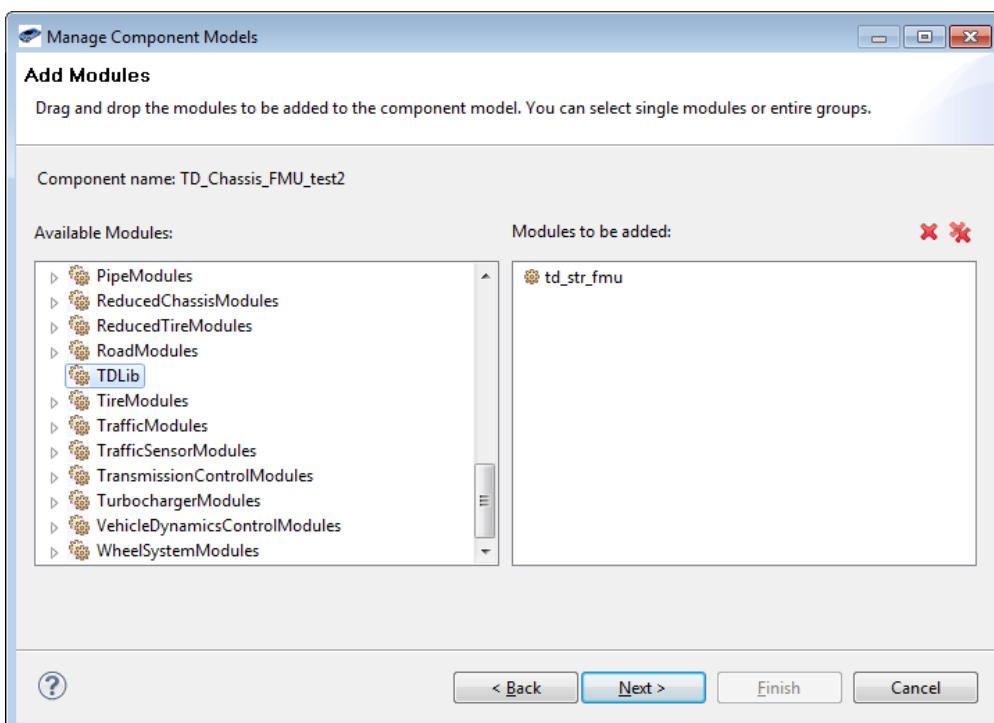
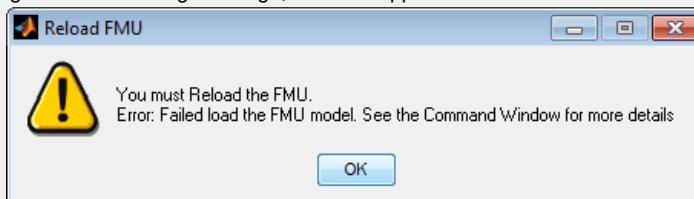


Figure: Add FMU Modules page

Notes:

- Ignore the following message, should it appear:



Importing FMU Components into Simulation Models

The objective of this tutorial step is to show how to import FMU components into simulation models in DYNA4. The following topics are addressed:

- Import newly created FMU component into the DYNA4 framework
- Create a module parameter file

Proceed as explained in the following:

1. As a final step import the newly created FMU component from the DYNAdatabase into the actual simulation model (see figure Replace FMU components dialog).

For details refer to Replacing Model Components in the DYNA4 User Manual.

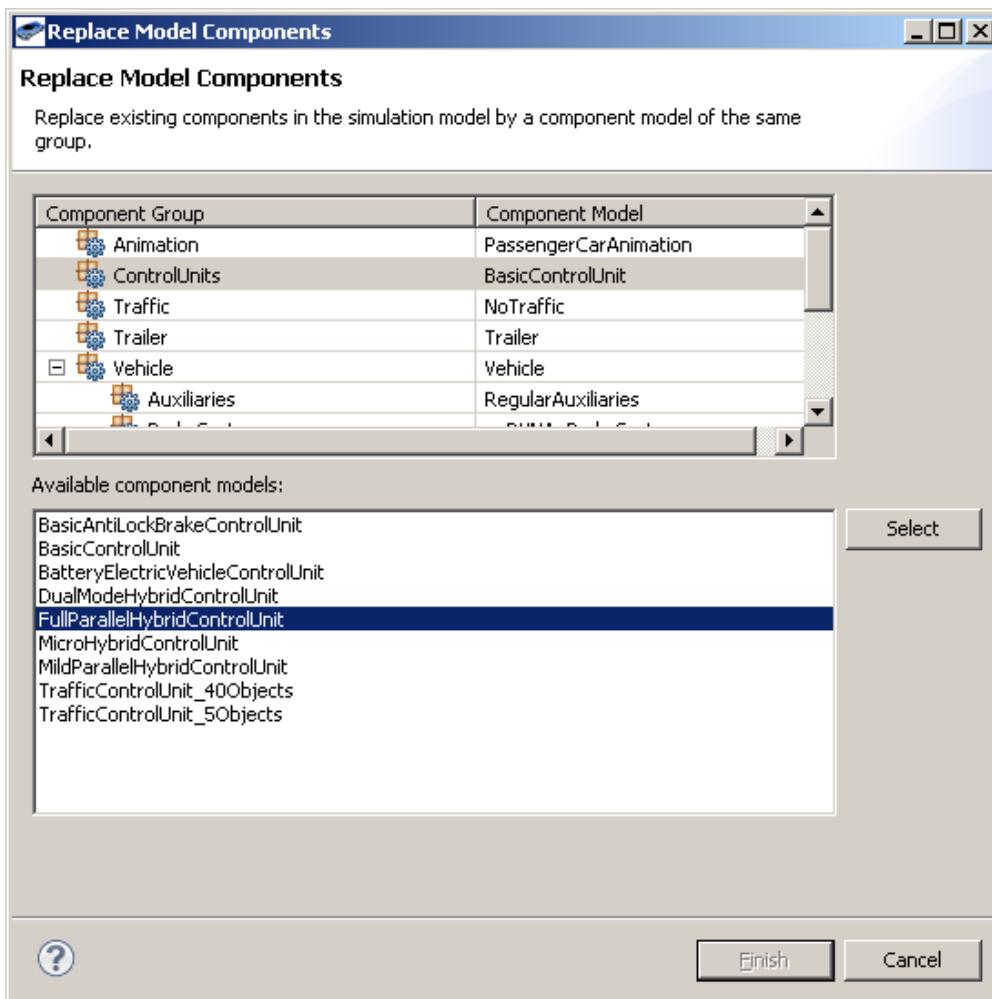


Figure: Replace FMU components dialog

2. Create a module parameter file.

Now that the FMU component and the corresponding user-defined FMU module are included in the active simulation model, the data can be applied directly from the DYNA4 framework into the MATLAB workspace.

Once the FMU component has been selected, a parameter file can be created and loaded into the active data set. If the module is new, there will be no existing data file, a new one can be created (see figure Create new module parameter file).

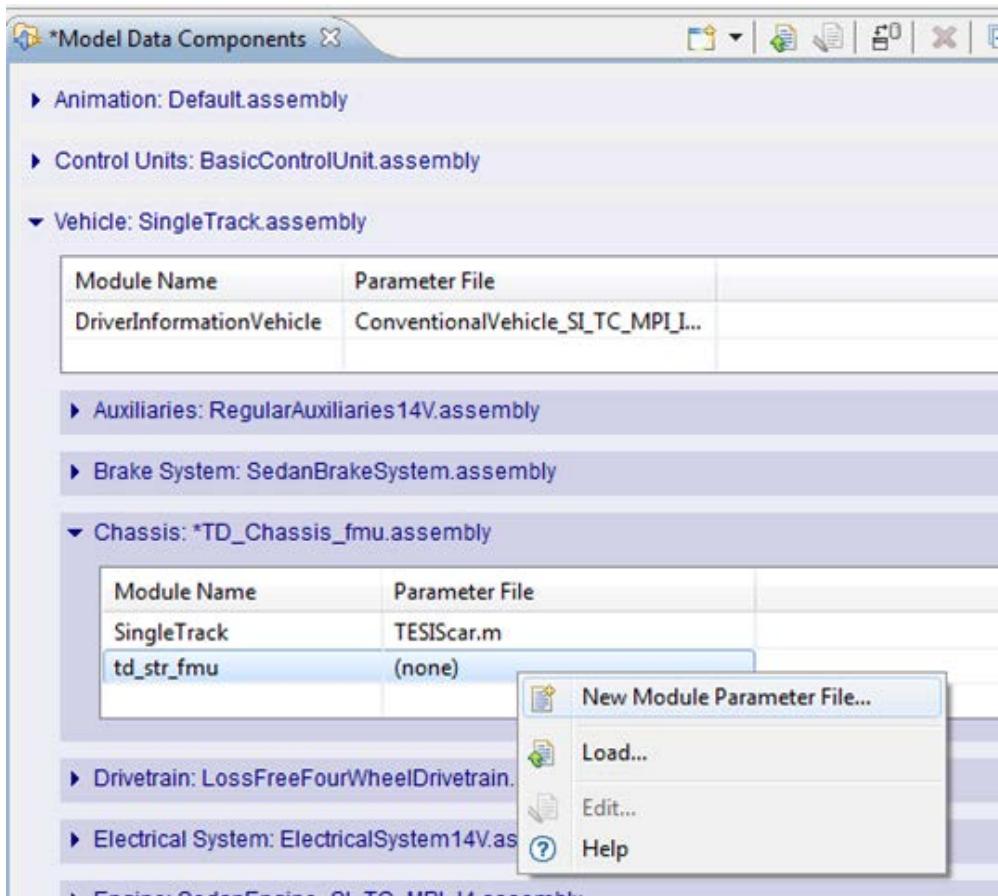


Figure: Create new module parameter file

The parameters for the model can be entered via the GUI when one double clicks on the desired mfile. Upon saving the mfile, the parameters for the FMU will be applied to the MATLAB workspace in the MDL structure.

The use of the DYNA4 parameter file is optional and depending on the desired data apply method used, the parameter file may be empty. The options regarding data apply are discussed in more detail in [Applying Data to FMU Blocks](#).

Working with FMUs in DYNA4

The objective of this tutorial step is to show how to work with FMUs in the DYNA4 simulation framework after the import steps have been executed successfully. The following topics are addressed:

- Applying Data to FMU Blocks: Depicts options for applying data to the FMU block
- Tracing Signals of FMU Blocks: Describes options for tracing signals from the FMU block
- Simulating with FMU Blocks: Shows tips regarding the simulation of FMUs in the Simulink environment using DYNA4

Applying Data to FMU Blocks

The objective of this tutorial step is to show how to apply data to FMU blocks. Following topics are covered:

- How to assign DYNA4 module parameters to FMU block parameters.

Notes on general FMI Toolbox restrictions:

- Any tables used as parameters in the FMI model must be of a fixed size. e.g. the parameters may vary but the table dimensions must remain constant.

Once the FMU is part of the active simulation model, it is possible to assign DYNA4 parameters to the corresponding FMU parameters.

Proceed as explained in the following

1. Open the DYNA4 module library in Simulink
2. Select "View Mask" in the module and go to the **DYNA4 Module Parameters** pane. There you can see all available DYNA4 parameters for this module.

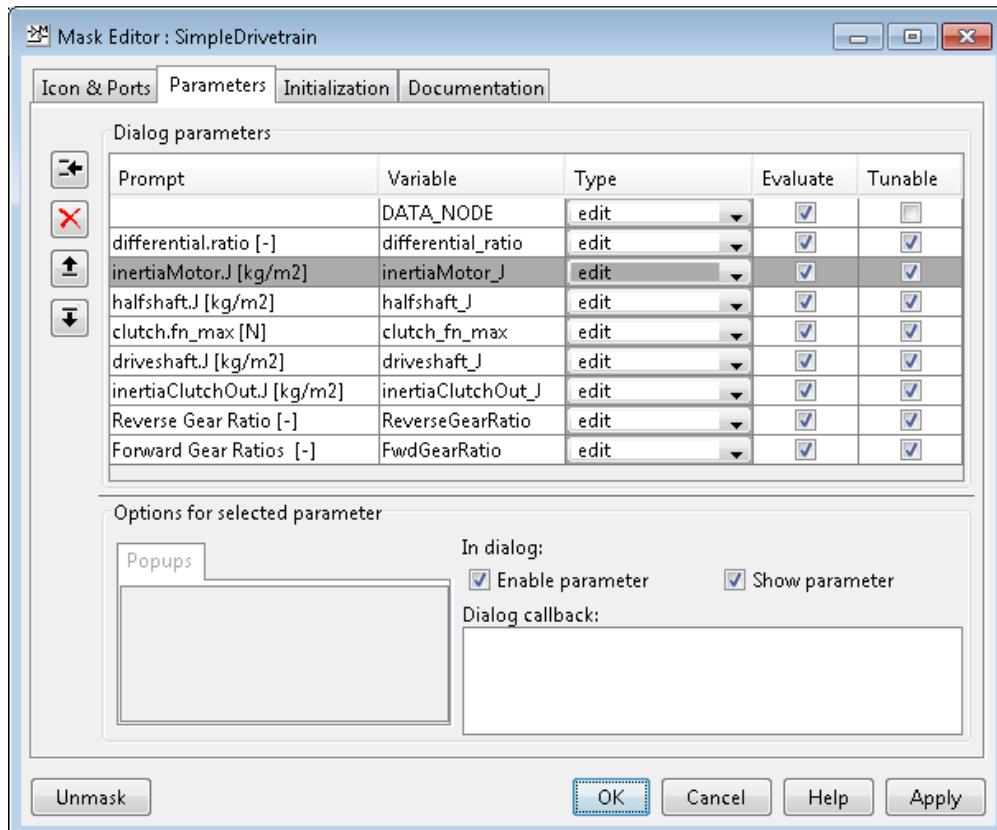


Figure: DYNA4 Module Parameters

3. Open the FMU Edit Mask (double click on the FMU block) and go to the "Parameters & start values" pane.

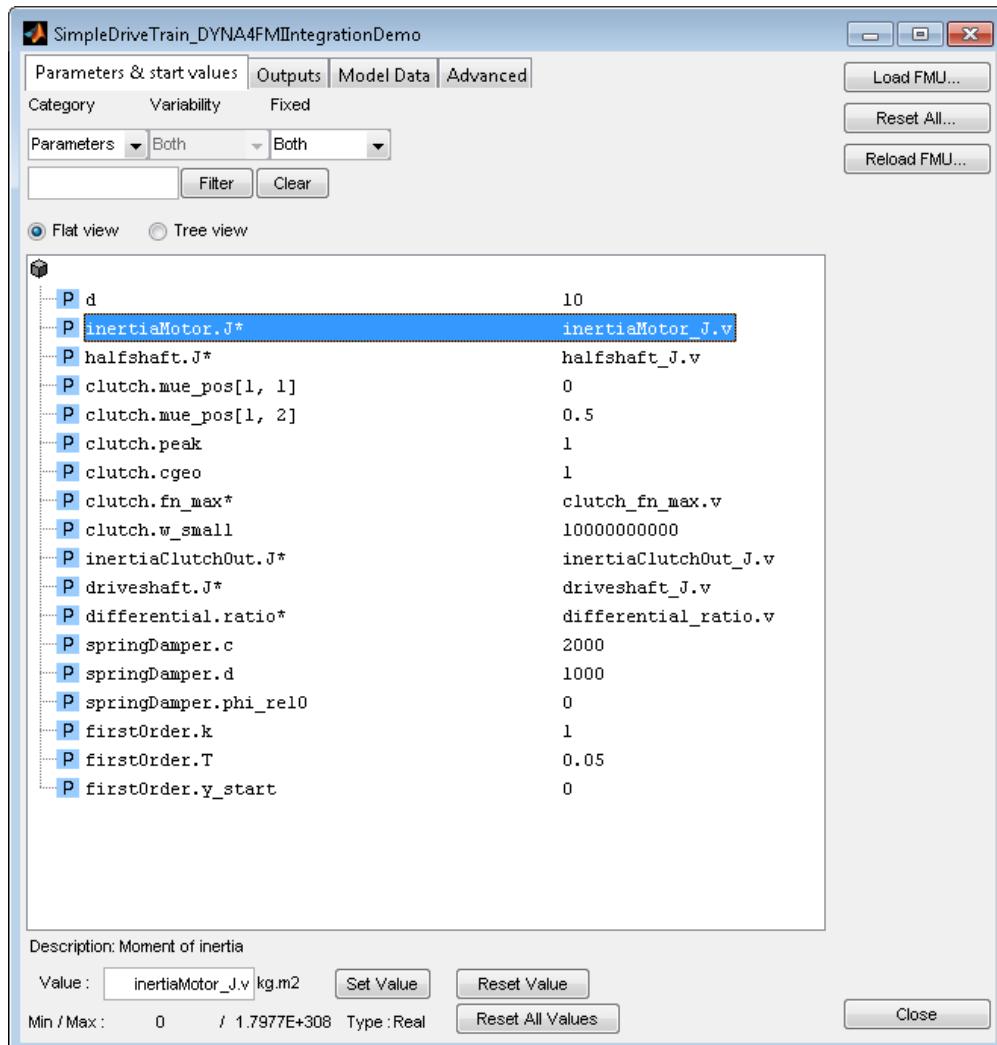


Figure: FMU Edit Mask

4. For each FMU parameter you want to change in DYNA4: Enter the corresponding DYNA4 parameter name in the Value field in the FMU edit mask and press „Set Value“.

Notes:

- It is important to add a .v suffix to the DYNA4 parameter name as this specifies the used parameter value structure element. (see [Figure FMU edit mask](#))

5. Close the FMU edit mask.

6. Save the DYNA4 module library.

From now on the selected DYNA4 module parameters are automatically assigned to the corresponding FMU parameters during the DYNA4 apply procedure.

Tracing Signals of FMU Blocks

The objective of this tutorial step is to show different options how to trace signals of an FMU block

in DYNA4. The following topics are addressed:

- Using the DYNA4 trace mechanism
- Using the FMI Toolbox report generation function

DYNA4 trace mechanism

It is possible to record individual signals or complete Simulink buses using the DYNA4 Trace mechanism. For further details please refer to Recording a User-Defined Simulink Signal and Recording a complete Simulink Bus in the DYNA4 User Manual.

This workflow requires that all desired signals are defined as outputs from the FMU block. For further details on how to specify additional outputs for the FMU block please refer to http://www.modelon.com/fileadmin/user_upload/Products/Modelon/FMIT/FMIT-1_7-UsersGuide.pdf.

Trace functionality of the FMI Toolbox for MATLAB

The alternative is to use the trace functionality of the FMI Toolbox, which traces the internal signals in the FMU and creates a separate result file.

The result file can then be processed using MATLAB functions from the Modelon FMI Toolbox. For further details please refer to http://www.modelon.com/fileadmin/user_upload/Products/Modelon/FMIT/FMIT-1_7-UsersGuide.pdf.

It is also possible to integrate these results into a DYNA4 postprocessing function and combine them with the trace results provided by DYNA4. For further information on how to integrate user functions into the DYNA4 task concept please refer to Using a Function Node in the DYNA4 User Manual.

Simulating with FMU Blocks

Some general notes regarding simulation with the FMU block and DYNA4:

- FMI provides a standardized format which allows the easy transfer of models between tools. It should however be considered that numerical problems can arise if this integration is not done in the most appropriate manner. This applies of course generally for the simulation of FMUs in Simulink, and is not limited specifically to DYNA4 models.
- Most DYNA4 models are (currently) restricted to operate with the ODE1 Solver in Simulink. Therefore use of the FMU block for Model Exchange (FMU ME) is restricted to this solver which means that no variable step size solver or higher order solvers are supported.
- The use of the ODE1 solver may not be an appropriate choice for some complex physical models. It may be necessary to use a very small step size in order to achieve good results, at the cost of performance.
- Using the FMU block for Co-simulation (FMU CS) removes this restriction on the solver type for the FMU. When exported in the CS format, the FMU will use its own solver, which is to be defined when the FMU is exported.
- Simulating with an FMU model that has been exported in the CS Format requires that the host computer has access to a license from the exporting tool.
- Using the FMI Toolbox for Matlab from Modelon and only providing DLL binaries for the FMU, only the simulation modes "Normal" and "Accelerator" can be used in Simulink.